11-3-2019

Rating mechanisms for sustainability of crowdsourcing platforms

Chenxi Qiu
Rowan University, qiu@rowan.edu

Anna Squicciarini

Sarah Rajtmajer

Rating Mechanisms for Sustainability of Crowdsourcing Platforms

Chenxi Qiu
qiu@rowan.edu
Department of Computer Science
Rowan University
Glassboro, New Jersey

Anna Squicciarini and Sarah Rajtmajer
{acs20,smr48}@psu.edu
College of Information Science and Technology
Pennsylvania State University
University Park, Pennsylvania

ABSTRACT

Crowdsourcing leverages the diverse skill sets of large collections of individual contributors to solve problems and execute projects, where contributors may vary significantly in experience, expertise, and interest in completing tasks. Hence, to ensure the satisfaction of its task requesters, most existing crowdsourcing platforms focus primarily on supervising contributors' behavior. This lopsided approach to supervision negatively impacts contributor engagement and platform sustainability.

In this paper, we introduce rating mechanisms to evaluate requesters' behavior, such that the health and sustainability of crowdsourcing platform can be improved. We build a game theoretical model to systematically account for the different goals of requesters, contributors, and platform, and their interactions. On the basis of this model, we focus on a specific application, in which we aim to design a rating policy that incentivizes requesters to engage less-experienced contributors. Considering the hardness of the problem, we develop a time efficient heuristic algorithm with theoretical bound analysis. Finally, we conduct a user study in Amazon Mechanical Turk (MTurk) to validate the central hypothesis of the model. We provide a simulation based on 3 million task records extracted from MTurk demonstrating that our rating policy can appreciably motivate requesters to hire less-experienced contributors.

CCS CONCEPTS

• **Social and professional topics**; • **Theory of computation** → *Mathematical optimization*; • **Computing methodologies** → *Modeling and simulation*;

KEYWORDS

Crowdsourcing; rating policy; bi-level programming

ACM Reference Format:
Chenxi Qiu and Anna Squicciarini and Sarah Rajtmajer. 2019. Rating Mechanisms for Sustainability of Crowdsourcing Platforms . In *The 28th ACM International Conference on Information and Knowledge Management (CIKM '19), November 3–7, 2019, Beijing, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3357384.3357933

CIKM '19, November 3–7, 2019, Beijing, China

ACM ISBN 978-1-4503-6976-3/19/11…$15.00
https://doi.org/10.1145/3357384.3357933

1 INTRODUCTION

Crowdsourcing systems enable new types of problem solving and creative production across a range of domains, from citizen science [13] to product development [32], entertainment and the arts [28]. Individuals involved in crowdsourcing are typically heterogeneous, self-selected, and voluntary participants engaged in temporary, decentralized problem-solving activities, for a number of selfish and altruistic reasons [18]. In particular, these systems offer individuals opportunities for learning, social mobility and global economy, while enabling work that may be too difficult for machines, or too large a scale for small groups [7].

The success of crowdsourcing platforms depends on accessibility to *high quality* work despite its inherent, fundamental challenges: individual contributors are often distant; maintain some degree of anonymity; are not thoroughly vetted or overseen; and work for relatively short-term rewards [35, 39]. Accordingly, thoughtful design of *incentivization schemes* is critically important. A number of recent studies have demonstrated the power of incentive schemes [18, 36, 37] to elicit fair and accurate ratings and productivity in crowdsourcing, particularly when most aspects of performance are measurable. However, these approaches tend to favor task requesters over contributors, in that they are designed to enhance contributors' productivity and throughput. For instance, in platforms offering micro-tasks, several incentive schemes (e.g., [18, 37]) provide higher compensation to contributors with positive historical records. Although intuitive, these approaches may eventually erode the platform's sustainability over time, as they discourage contributors with less experience from participating (and therefore gaining experience). Moreover, some requesters may take the advantage of the platform policies by rejecting work once completed [30].

In this paper, we design incentive mechanisms to reduce the information asymmetry between requesters and contributors, such that the health of a crowdsourcing platform can be maintained in the long run. Specifically, we provide a sound approach to rate *requesters* based on their *compensation policies*. Ratings are calculated based on the policies' positive impact on platform sustainability, i.e., increasing number of contributors. Each requester's rating is visible to the public, and in particular, to potential contributors. As such, requesters are motivated to obtain higher rating to attract qualified contributors, which in turn bolsters platform sustainability.

It is worth noting that designing such rating policy is non-trivial considering the conflicting and competing objectives of different stakeholders within a platform. In this paper, we describe behaviors of the platform, requesters, and contributors by assuming a three-layer architecture, where the platform (in the top layer) rates each requester (in the second layer), and each requester in turn

compensates their contributors (in the third layer) according to contributors' performance. We model between-layer interactions as a Stackelberg game, a strategic game where a leader makes decision first and followers move sequentially:
1) *Platform vs. requesters*, where the platform (as the leader) publishes their rating policy to requesters, and then requesters (as followers) specify their compensation policy.
2) *Requesters vs. contributors*, where requesters (as leaders) first determine their compensation policy, and then contributors (as followers) decide the effort level to complete tasks.

On the basis of this game theoretical model, we formally formulate the *optimal rating policy (ORP)* problem as a three-level programming problem, where optimization related to the contributors' effort level is taken as a constraint when requesters determine the compensation policy, and the optimization related to the requesters' compensation policy is taken as a constraint when calculating the optimal rating policy for the platform.

Given this optimization framework, we then target a specific scenario, in which the sustainability goal from the platform is to increase the participation of "new" contributors, i.e., contributors with less experience/lower skill levels. In this scenario, the platform is responsible not only for delivering high quality work to task requesters, but also for attracting and training new contributors with diverse background to secure its own long-term sustainability. Considering the hardness of ORP, we develop a time-efficient algorithm by resorting to optimization techniques including, *approximation*, *level reduction*, and *relaxation*. For theoretical interest, we also derive an upper bound on platform utility and compare the closeness of results derived by our algorithm to optimal.

Finally, we conduct a user study to validate the central hypothesis of our model, namely the ability of a rating scheme to appreciably impact requester behavior. We involve MTurk contributors, asking them to provide insights on the design of specific tasks and their willingness to support the inclusion of contributors with less experience. We also carry out an extensive evaluation of our proposed rating policy by leveraging a real dataset (over 3 million task records) extracted from MTurk. The experimental results demonstrate that our approach can significantly improve the overall performance of less-experienced contributors, 214.3% higher than their recorded performance in the dataset.

We summarize our contributions as follows:
1) We first build a general game theoretical model that takes into account different objectives of platform, requesters, and contributors, and their interactions. Based on the model, we formulate the optimal rating policy problem in crowdsourcing, a new class of optimization problems that aim to maximize platform sustainability.
2) Based on the general optimization framework, we target a specific scenario wherein the platform aims to maximize the participation from new contributors. As a solution, we propose a time-efficient algorithm with theoretical bound provided.
3) We conduct an extensive evaluation of the framework, using real-data, simulations and through a proof of concept user study. The latter helps us validate the impact of rating policy to requesters' decision. The trace-driven simulation (that relies on the real dataset) demonstrates the superiority of our rating policy in terms of engaging less-experienced contributors in crowdsourcing services.

The remainder of the paper is organized as follows: In the next section, we present our general model and problem formulation. In Section 3, we focus on a specific scenario and develop the algorithm to derive the rating policy. In Section 4, we present the results of our user study and evaluate the performance of our rating policy with data-driven simulations. Finally, we present related work in Section 5 and conclude in Section 6.

2 MODEL AND PROBLEM FORMULATION

Informally, we focus on a three-party scheme, where requesters offer microtasks for contributors to complete. Microtasks are made available by a platform provider, that acts as a host and supervisor of the ongoing crowdwork transactions. We assume that contributors select preferred microtasks as they are made available. Workers can leave and join the platform at any time, but are typically persistent and are in a larger number than requesters. Requesters aim to "hire" contributors who can provide quality responses/work in a timely fashion. In this section, we formalize this model, including notations and assumptions of *rating and compensation policies* (in Section 2.1) and *objectives of different stakeholders* (in Section 2.2). Based on the model, we then formulate the *optimal rating policy* problem (in Section 2.3). Table 1 lists the main notations as well as their explanations that will be used in this paper.

Table 1: Notations and descriptions

Notation	Description
M	The number of requesters
N	The number of contributors
$x_{i,j}$	Contributor i's effort level on the tasks from requester j
$\mathcal{X}$	Effort level space
$\mathbf{b}_i$	Contributor i's background vector
$\mathbf{e}_j$	Requester j's task feature vector
$q_{i,j}$	Contributor i's work quality on the tasks from requester j
$c_{i,j}$	Contributor i's compensation paid by requester j
C	Compensation space
$\mathbf{y}_j$	Parameter vector of requester j's compensation policy
$\mathcal{Y}$	Compensation policy space
f	Compensation function
h	Rating function
$\mathcal{R}$	Rating space
$\mathcal{I}$	The set of polyhedrons (cells) partitioned in the compensation policy space
I_l	The lth polyhedron (cell) in $\mathcal{I}$
z_l	The rating assigned to the lth polyhedron I_l
α_i^{laz}	Contributor i's laziness coefficient
α_j^{cost}	Requester j's cost coefficient
α_j^{rate}	Requester j's rating coefficient
g	Platform's utility function

2.1 Compensation and Rating Policies

Compensation policy. In practical terms, compensation policies calculate optimal compensation strategies for contributors, considering contributors' expected effort level (e.g., average working time/task), background (e.g., skill level and expertise), and tasks' features (e.g., difficulty levels).

We consider a scenario composed of M requesters $\{1, 2, ..., M\}$ and N contributors $\{1, 2, ..., N\}$. Each requester j posts a set of tasks on a crowdsourcing platform with a *compensation policy* displayed, and then contributors determine whether to complete the tasks and how much effort to put in, in order to receive compensation. We use

the variable $x_{i,j} \in \mathcal{X}$ to represent each contributor i's effort level on requester j's tasks, where $\mathcal{X} \subset \mathbb{R}$ represents contributors' *effort level space*. Here, we normalize $x_{i,j}$ to the interval $[0,1]$ ($\mathcal{X} = [0,1]$). We use vectors $\mathbf{b}_i = \left[b_{i,1}, ..., b_{i,U}\right] \in \mathbb{R}^U$ and $\mathbf{e}_j = \left[e_{i,1}, ..., e_{i,T}\right] \in \mathbb{R}^T$ to represent the background of contributor i and the task feature of requester j. Given the above notations, the quality of work output by contributor i, denoted by $q_{i,j}$, can be represented as a function of his/her effort level $x_{i,j}$, given the *background* $\mathbf{b}_i$ and the task feature $\mathbf{e}_j$: $q_{i,j} = \xi\left(x_{i,j}; \mathbf{b}_i, \mathbf{e}_j\right)$.

We assume that the compensation that contributor i obtains from requester j, denoted by $c_{i,j} \in C$, depends on contributors' effort, background, and tasks' feature, where C denotes the *compensation space*. We can then represent requester j's compensation policy as a map $f : \mathcal{X} \mapsto C$. Therefore, contributor i's compensation $c_{i,j}$ is given by

$$c_{i,j} = f\left(x_{i,j}; \mathbf{b}_i, \mathbf{e}_j, \mathbf{y}_j\right) \tag{1}$$

where vector $\mathbf{y}_j = \left[y_{j,1}, ..., y_{j,L}\right]$ includes the parameters to characterize f (a detailed example will be introduced in Section 3).

Property 2.1. *f is assumed to have the following properties:*

P-I: *Contributor i obtains no payment if his/her effort level is 0, i.e., $f\left(0, \mathbf{b}_i, \mathbf{e}_j; \mathbf{y}_j\right) = 0$.*

P-II: *$f\left(x_{i,j}; \mathbf{b}_i, \mathbf{e}_j, \mathbf{y}_j\right)$ is monotonically increasing over $x_{i,j} \in [0,1]$.*

Here, we allow requester j to determine his/her compensation policy by specifying $\mathbf{y}_j$, and hence $\mathbf{y}_j$ is considered as the *decision variables* of requester j. We let $\mathcal{Y} \subseteq \mathbb{R}^L$ represent the set of all possible decisions made by requesters, namely the *compensation policy space*. Hence, each $\mathbf{y}_j \in \mathcal{Y}$. We assume $\mathcal{X}$, C, and $\mathcal{Y}$ are all *compact space*.

Rating policy. We assume that the platform rates each requester j according to requester j's decision $\mathbf{y}_j$ that specifies the compensation policy f. For simplicity, we assume that other factors affecting the quality of a requester are ignored or equal across requesters, i.e. all tasks are equally well designed, and requesters are all honest, and follow the published compensation policies.

Therefore, the rating policy, denoted by h, can be modeled as a map from the *compensation policy* space $\mathcal{Y}$ to the *rating* space $\mathcal{R}$, i.e., $h : \mathcal{Y} \mapsto \mathcal{R}$, where $\mathcal{R} \subset \mathbb{R}$ is *compact*. To derive the optimal rating function using optimization techniques, it is necessary to describe the function by a finite set of decision variables. As such, we approximate the rating function h by

1) partitioning $\mathcal{Y}$ into a set of L-dimensional polyhedrons $\mathcal{I} = \left\{I_1, ..., I_l, ..., I_Q\right\}$;
2) assuming that the rating that requester j obtains is a constant z_l within each polyhedron I_l, where z_l is the decision made by the platform.

Hence, the rating policy h can be represented by a Q-dimensional step function:

$$h\left(\mathbf{y}_j; \mathbf{z}\right) = z_l, \text{ if } \mathbf{y}_j \in I_l, (l = 1, ..., Q), \tag{2}$$

where $\mathbf{z} = \left[z_1, ..., z_Q\right]$.

2.2 Objectives of Different Stakeholders

Contributor: Given the requester compensation policy f, each *contributor i* seeks to determine the effort level $x_{i,j}$ such that his/her compensation $f\left(x_{i,j}; \mathbf{b}_i, \mathbf{e}_j, \mathbf{y}_j\right)$ is maximized while the effort level $x_{i,j}$ is minimized:

$$\max \ f\left(x_{i,j}; \mathbf{b}_i, \mathbf{e}_j, \mathbf{y}_j\right) - \alpha_i^{\text{laz}} x_{i,j} \qquad \text{s.t. } x_{i,j} \in \mathcal{X} \tag{3}$$

where α_i^{laz}, called contributor i's *laziness coefficient*, represents contributor i's unwillingness to put effort in submitting correct tasks, i.e., higher α_i^{laz} implies less effort to be taken by the contributor.

Requester: Each requester j aims to determine $\mathbf{y}_j$ to *maximize the overall quality of contributors' ouput* $\sum_i \xi\left(x_{i,j}; \mathbf{b}_i, \mathbf{e}_j\right)$ and to *minimize the total cost* $\sum_i f\left(x_{i,j}; \mathbf{b}_i, \mathbf{e}_j, \mathbf{y}_j\right)$. Moreover, as contributors tend to choose completing tasks from requesters with higher rating, requester j also aims to improve his/her rating to compete with other requesters. Hence, the objective function of requester j can be represented by

$$\max \quad u_j^{\text{req}} = \sum_i \xi\left(x_{i,j}; \mathbf{b}_i, \mathbf{e}_j\right) - \alpha_j^{\text{cost}} \sum_i f\left(x_{i,j}; \mathbf{b}_i, \mathbf{e}_j, \mathbf{y}_j\right) + \alpha_j^{\text{rate}} h\left(\mathbf{y}_j; \mathbf{z}\right) \tag{4}$$

$$\text{s.t.} \quad \mathbf{y}_j \in \mathcal{Y}, \tag{5}$$

where the *cost coefficient* α_j^{cost} reflects the requester j's unwillingness to compensate contributors (i.e., higher α_j^{cost} implies lower compensation paid to contributors) and the *rating coefficient* α_j^{rate} indicates the requester's willingness to improve his rating (i.e., higher α_j^{rate} implies higher willingness to improve the rating).

Platform: We represent platform health and sustainability as a function of decisions from all the contributors ($\mathbf{X} = \left\{x_{i,j}\right\}_{N \times M}$) and requesters ($\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_M]$), and the rating policy ($\mathbf{z}$), given the all contributors' background information ($\mathbf{B} = [\mathbf{b}_1, ..., \mathbf{b}_N]$) and the features of all tasks from requesters ($\mathbf{E} = [\mathbf{e}_1, ..., \mathbf{e}_M]$):

$$\text{sustainability} = g\left(\mathbf{X}, \mathbf{Y}, \mathbf{z}; \mathbf{B}, \mathbf{E}\right). \tag{6}$$

We give a simple example of g here:

Example: As the success of crowdsourcing is highly due to the high number of contributors available in the system, it is of great importance to attract "new contributors" to participate. In this case, we could define g as the sum of new contributors' contributions to a given set of tasks. More precisely, suppose $b_{i,1}$ represents the experience of each contributor i (i.e., could be measured by the number of tasks contributor i has completed) since joining in the system, then g can be defined as

$$g\left(\mathbf{X}, \mathbf{Y}, \mathbf{z}; \mathbf{B}, \mathbf{E}\right) = \sum_{b_{i,1} \le \eta} \xi\left(x_{i,j}; \mathbf{b}_i, \mathbf{e}_j\right) \tag{7}$$

where η is a predefined threshold for $b_{i,1}$ to identify whether contributor i is "new" or not.

Besides the above example, we also support other objectives that benefit the platform health and sustainability such as skill acquisition, increased knowledge, upward mobility etc. Here, we can include different types of goals in the problem formulation and take a linear combination of their defined utility as the objective function to maximize. In what follows, we still use the general function $g\left(\mathbf{X}, \mathbf{Y}, \mathbf{z}; \mathbf{B}, \mathbf{E}\right)$ to represent the platform's utility. Formally, besides *requesters* and *contributors*, we add the third role in the system: the *platform*, that aims to maximize its own health and sustainability:

$$\max \quad g\left(\mathbf{X}, \mathbf{Y}, \mathbf{z}; \mathbf{B}, \mathbf{E}\right) \tag{8}$$

$$\text{s.t.} \quad \mathbf{z} \in \mathcal{Z}. \tag{9}$$

2.3 Problem Formulation.

We model the interaction between the requester and his/her contributors as a *Stackelberg game*, where a leader makes a decision (outlines the decision space) first and his/her followers move sequentially. More precisely, the requester (as the *leader*) first specifies the compensation policy, and the contributors (as *followers*) determine how much effort to put in.

Similarly, the interaction between the platform and requesters can be also modeled as a Stackelberg game, where the platform is taken as the leader who first publishes the rating policy and requesters are taken as followers who then specify their compensation policy according to the rating policy.

Considering the above two types of interactions, i.e., platform (layer 1) vs. requesters (layer 2), and requesters (layer 2) vs. contributors (layer 3), the *optimal rating policy (ORP)* problem can be formulated as a three-level optimization problem:

$$
\begin{aligned}
&\max\ g(\mathbf{X},\mathbf{Y},\mathbf{z};\mathbf{B},\mathbf{E}) \ \text{(Layer 1)}\\
&\text{s.t.}\\
&\quad \max\ \textstyle\sum_i \xi\left(x_{i,j};\mathbf{b}_i,\mathbf{e}_j\right) - \alpha_j^{\text{cost}} \sum_i f\left(x_{i,j};\mathbf{b}_i,\mathbf{e}_j,\mathbf{y}_j\right)\\
&\quad\quad +\alpha_j^{\text{rate}} h\left(\mathbf{y}_j;\mathbf{z}\right) \ \text{(Layer 2)}\\
&\quad \text{s.t.}\\
&\quad\quad \max\ f\left(x_{i,j};\mathbf{b}_i,\mathbf{e}_j,\mathbf{y}_j\right) - \alpha_i^{\text{laz}} x_{i,j} \ \text{(Layer 3)}\\
&\quad\quad \text{s.t.}\ \ x_{i,j}\in\mathcal{X},\ i=1,\ldots,N,\\
&\quad \mathbf{y}_j\in\mathcal{Y},\ j=1,\ldots,M,\\
&\mathbf{z}\in\mathcal{Z}
\end{aligned}
$$

The decision variables of the optimization problem in layer 1, 2, and 3 are $\mathbf{z}$, $\mathbf{y}_j$, and $x_{i,j}$ ($i=1,\ldots,N$, $j=1,\ldots,M$), respectively. The above hierarchical relationship results from the fact that the optimization related to the contributors' behavior is taken as a constraint when the requester makes the decision, and similarly, the optimization related to the requesters' behavior is taken as a constraint for the platform's decision. The objective is to find the optimal rating policy $\mathbf{z}$ for the platform to maximize a pre-defined dimension of system health and sustainability.

Like most works in the area of bi-level programming (BiP), in this paper, we assume the existence of *optimistic* bi-level optimum, where the followers are expected to choose a solution that is a best one from the point-of-view of the leader [5, 16, 27, 34]. The solution of the above problem is provided as the suggested rating policy to the platform.

3 RATING POLICY TO INCENTIVIZE CONTRIBUTORS WITH LESS EXPERIENCE

In this section, we concentrate on a scenario where the platform aims to incentivize contributors with less experience to actively participate in crowdsourcing services. In Section 3.1, by considering contributors' experience as a main factor for requesters to determine the compensation, we assume compensation policies in a specific class. Based on this assumption, we then propose a time efficient solution to derive the optimal rating policy in Section 3.2&3.3. Table 2 lists the additional notations that will be used in this section.

3.1 Assumptions of Compensation Policies and Platform Utility

Compensation policies. In crowdsourcing systems such as MTurk [31], requesters may have requirements for contributors' experience (e.g., represented as accumulated rating) when distributing tasks. We model each requester j's experience requirement as a parameter in its compensation policy f, denoted by $y_{j,2}$. Without loss of generality, we let $b_{i,1}$ denote contributor i's experience. Therefore, contributor's work can be accepted by requester j if only if $b_{i,1}\geq y_{j,2}$. In requester j's compensation policy, we also consider another parameter $y_{j,1}$, which reflects the average compensation awarded to all contributors. Like [36], we assume f to be quadratic and continuous in its domain $\mathcal{X}$.

Table 2: Notations and descriptions (Section 3)

Notation	Description
$y_{j,1}$, $y_{j,2}$	Compensation policy parameters: $y_{j,1}$ reflects requester j's overall compensation awarded to all contributors; $y_{j,2}$ denotes the requester's minimum requirement for contributors' experience
$b_{i,1}$	Contributor i's experience
Θ	Heaviside step function
γ^{new}	Weight assigned to new contributors' performance in platform utility
$\mathbf{Z}=\{z_{l,k}\}$	$\mathbf{Z}$ is the rating matrix. $z_{l,k}$ represents the rating assigned to cell $I_{l,k}$
$u_j^{\text{req}}(I)$	The utility of requester j given $[y_{j,1}, y_{j,2}]\in I$
η	Threshold to identify "new" contributors

According to the above assumptions and Property 2.1 (for general compensation policies), we can represent f as:

$$f\left(x_{i,j};\mathbf{b}_i,\mathbf{e}_j,\mathbf{y}_j\right)=\Theta(b_{i,1}-y_{j,2})y_{j,1}x_{i,j}(2-x_{i,j}) \tag{10}$$

where Θ is the *Heaviside step function* [26]:

$$\Theta(w)=\begin{cases}1 & w\geq 0\\ 0 & \text{otherwise}\end{cases} \tag{11}$$

indicating that contributor i's response will be accepted if only if his/her experience $b_{i,1}$ is higher than the requirement $y_{j,2}$.

According to Equation (10), each requester j determines his/her compensation policy by specifying the two parameters $\mathbf{y}_j=\left[y_{j,1},y_{j,2}\right]$. In this case, $\mathcal{Y}\subset\mathbb{R}^2$. We normalize both $y_{j,1}$ and $y_{j,2}$ to $[0,1]$.

Platform's utility. We make the following two assumptions regarding the utility function g of the platform:
A1) g increases monotonically with the increase of contributors' overall performance;
A2) g increases monotonically with the increase of the overall performance of "new" contributors, whose experience $b_{i,1}$ is no higher than a predefined threshold η;

According to A1 and A2, we define g as follows:

$$
\begin{aligned}
&g(\mathbf{X},\mathbf{Y};\mathbf{B},\mathbf{E}) \qquad\qquad (12)\\
=\ &\underbrace{\sum_j\sum_i \xi\left(x_{i,j};\mathbf{b}_i,\mathbf{e}_j\right)}_{\text{contributors' overall performance}} + \gamma^{\text{new}} \underbrace{\sum_j \sum_{b_{i,1}\leq\eta} \xi\left(x_{i,j};\mathbf{b}_i,\mathbf{e}_j\right)}_{\text{new contributors' performance}}
\end{aligned}
$$

where γ^{new} is a coefficient to reflect how much the platform focuses on motivating new contributors' participation.

Rating policy representation. We now determine the approximated representation of the rating policy h in this specific scenario: We 1) partition $\mathcal{Y}$ into a 2-dimensional grid $\mathcal{I} = \{I_{l,k}\}_{L\times K}$, as Figure 1 shows, where each cell

$$I_{l,k} = \left(\frac{l-1}{L}, \frac{l}{L}\right] \times \left(\frac{k-1}{K}, \frac{k}{K}\right] (l = 1, ..., L, k = 1, ..., K), \quad (13)$$

2) assume that the rating $z_{l,k}$ assigned to requester j is a constant given $\mathbf{y}_j$ within each cell $I_{l,k}$.

Therefore, h can be written as a 2 dimensional step function:

$$h\left(\mathbf{y}_j; \mathbf{Z}\right) = z_{l,k}, \text{ if } \mathbf{y}_j \in I_{l,k}, (l = 1, ..., L, k = 1, ..., K) \quad (14)$$

where $\mathbf{Z} = \{z_{l,k}\}_{L\times K}$.

In addition, to incentivize requesters to hire contributors with less experience, we award requesters high rating if they have lower requirements for contributors' experience (Property 3.1(a)). Considering that requesters may pay contributors with unfairly low compensation [30], we penalize requesters with lower rating if their overall compensation is lower (Property 3.1(b)).

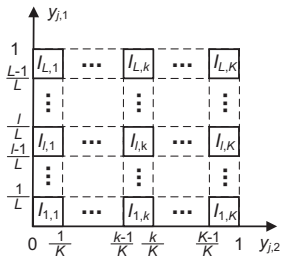


Figure 1: Grid of $\mathcal{Y}$.

Property 3.1. *(a) $h\left(\mathbf{y}_j; \mathbf{Z}\right)$ decreases monotonically with $y_{j,2}$, which is enforced by the constraints: $z_{l,k-1} \geq z_{l,k}$, for each $l = 1, ..., L$, $k = 2, ..., K$.*
(b) $h\left(\mathbf{y}_j; \mathbf{Z}\right)$ increases monotonically with $y_{j,1}$, which is enforced by the constraints: $z_{l-1,k} \leq z_{l,k}$, for each $l = 2, ..., L$, $k = 1, ..., K$.

3.2 Level Reduction of ORP

Even with additional assumptions made on requesters' compensation policies in this specific scenario, ORP is still hard to solve due to its three-level structure. Fortunately, the problem in *Layer 3* is convex and sufficiently regular [41], i.e., the objective function (defined by Equation (10) is quadratic and the feasible region is an interval (i.e., $x_{i,j} \in [0, 1]$). Hence, the problem can be replaced by a closed-form *equilibrium constraint*, as described in Proposition 3.1:

Proposition 3.1. *Given requester j's compensation policy coefficient $\mathbf{y}_j$, the optimal reaction of contributor i, denoted by $x^*_{i,j}$, is derived as*

$$x^*_{i,j} = \phi\left(\mathbf{y}_j; \alpha_i^{\text{laz}}\right) \quad (15)$$
$$= \begin{cases} 0 & y_{j,2} > e_{i,j} \text{ or } y_{j,1} \in \left[0, \frac{\alpha_i^{\text{laz}}}{2}\right) \\ \frac{2y_{j,1} - \alpha_i^{\text{laz}}}{y_{j,1}} & y_{j,2} \leq e_{i,j} \text{ and } y_{j,1} \in \left[\frac{\alpha_i^{\text{laz}}}{2}, \alpha_i^{\text{laz}}\right] \\ 1 & y_{j,2} \leq e_{i,j} \text{ and } y_{j,1} \in \left(\alpha_i^{\text{laz}}, \infty\right) \end{cases}$$

Proof. The detailed proof can be found in Appendix. □

Then, the optimization in *Layer 2&3* can be rewritten as:

$$\begin{array}{ll} \max & \sum_i \xi\left(x_{i,j}; \mathbf{b}_i, \mathbf{e}_j\right) - \alpha_j^{\text{cost}} \sum_i f\left(x_{i,j}; \mathbf{b}_i, \mathbf{e}_j, \mathbf{y}_j\right) \\ & +\alpha_j^{\text{rate}} h\left(\mathbf{y}_j; \mathbf{z}\right) \text{ (Layer 2)} \\ \text{s.t.} & x_{i,j} = \phi(\mathbf{y}_j; \alpha_i^{\text{laz}}),\ \forall x_{i,j} \in \mathcal{X} \\ & \mathbf{0} \leq \mathbf{y}_j \leq \mathbf{1},\ j = 1, ..., M, \end{array}$$

and ORP is reduced to a BiP.

Note that the optimization problem in Layer 2 is non-convex, and hence it is hard to apply level reduction by deriving the equilibrium constraints in closed form again.

3.3 Algorithm Design

In this section, we target addressing the BiP composed of Layer 1&2,. BiP in general is known to be strongly NP-hard [14], and it has been proven that merely evaluating a solution for optimality is also a NP-hard task [6, 42]. Therefore, in this section, we aim to design a time efficient heuristic that can achieve near-optimal.

3.3.1 Single requester case. We start with a simpler version of ORP, where only the interaction between the platform and a single requester j is considered. In this case, the platform aims to maximize the utility defined by

$$g_j\left(\mathbf{y}_j; \mathbf{B}, \mathbf{e}_j\right) \quad (16)$$
$$= \sum_i \xi\left(\phi\left(\mathbf{y}_j; \alpha_i^{\text{laz}}\right), \mathbf{b}_i, \mathbf{e}_j\right) + \gamma^{\text{new}} \sum_{b_{i,1} \leq \eta} \xi\left(\phi\left(\mathbf{y}_j; \alpha_i^{\text{laz}}\right), \mathbf{b}_i, \mathbf{e}_j\right).$$

This simplified ORP is actually composed of two single-level optimization problems **P1** and **P2** that can be solved sequentially:
1) We note that the platform utility defined in Equation (16) depends only on the requester j's decision $\mathbf{y}_j$. Hence, **P1** is defined to find $\mathbf{y}_j^*$ to maximize g_j:

$$\mathbf{y}_j^* = \arg\max_{\mathbf{y}_j \in \mathcal{Y}} g_j\left(\mathbf{y}_j; \mathbf{B}, \mathbf{e}_j\right). \quad (17)$$

which can be solved by well-developed methods such as the subgradient methods [17].
2) After deriving $\mathbf{y}_j^*$, in **P2**, we derive the rating policy $\mathbf{Z}_j^*$ such that requester j utility is maximized at I_j^*, i.e.,

$$u_j^{\text{req}}\left(I_j^*\right) - u_j^{\text{req}}(I) \geq \Delta,\ \forall I \quad (18)$$

where I_j^* is the cell $\mathbf{y}_j^*$ located in and $\Delta > 0$ is a constant. That is, with $\mathbf{Z}_j^*$, requester j will make his/her optimal decision in I_j^*, which is close to $\mathbf{y}_j^*$, and hence closely approach the maximum g_j. To ensure Equation (18), we derive the following constraints for the entries in $\mathbf{Z}_j^*$ according to Equation (4):

$$z_{I_j^*} - z_I \geq \psi_{I_j^*, I},\ \forall I \in \mathcal{I} \setminus I_j^* \quad (19)$$

where

$$\psi_{I_j^*, I} = \left(\alpha_j^{\text{cost}} \sum_i \left(f\left(\phi\left(I_j^*; \alpha_i^{\text{laz}}\right), \mathbf{b}_i, \mathbf{e}_j; I\right) - f\left(\phi\left(I; \alpha_i^{\text{laz}}\right), \mathbf{b}_i, \mathbf{e}_j; I\right)\right)\right.$$
$$\left. - \sum_i \left(\xi\left(\phi\left(I_j^*; \alpha_i^{\text{laz}}\right), \mathbf{b}_i, \mathbf{e}_j\right) - \xi\left(\phi\left(I; \alpha_i^{\text{laz}}\right), \mathbf{b}_i, \mathbf{e}_j\right)\right) + \Delta\right) \Big/ \alpha_j^{\text{rate}}$$

is a contant.

Moreover, to guarantee that the requester's reaction $\mathbf{y}_j$ converges to I_j^* instead of any local utility maxima, we require that *the requester's utility has no local maxima except I_j^* in* $\mathbf{Z}$. According

to the position of I_j^*, we partition the decision space $\mathcal{Y}$ into four regions I, II, III, and IV, as Figure 2(a) shows. We then determine the following constraints to ensure that requester j's decision can eventually flow to I_j^* no matter where its initially status is (as shown in Figure 2(b)):

$$\begin{cases} z_{l,k} - z_{I_{l,k+1}} \geq \psi_{I_{l,k},I_{l,k+1}} & \text{in regions I\&II} \\ z_{l,k} - z_{I_{l,k-1}} \geq \psi_{I_{l,k},I_{l,k-1}} & \text{in regions III\&IV} \\ z_{l,k} - z_{I_{l+1,k}} \geq \psi_{I_{l,k},I_{l+1,k}} & \text{in regions I\&IV} \\ z_{l,k} - z_{I_{l-1,k}} \geq \psi_{I_{l,k},I_{l-1,k}} & \text{in regions II\&III} \end{cases} \tag{20}$$

Equation (20) indicates that the requester's decision will flow from $I_{l,k+1}$ to $I_{l,k}$ in regions I&II, and flow from $I_{l,k-1}$ to $I_{l,k}$ in regions III&IV; similarly, the decision will flow from $I_{l+1,k}$ to $I_{l,k}$ in regions I&IV, and flow from $I_{l-1,k}$ to $I_{l,k}$ in regions III&IV. For simplicity, we use Ω_j to represent the set of entry pairs in $\mathbf{Z}$ restricted by the constraints in Equation (20).

3.3.2 Multi requester case. To derive the rating policy for multiple requesters, a possible solution is to apply the algorithm introduced in Section 3.3.1 directly to each requester j to obtain Ω_j ($j = 1, ..., M$). Then, by taking together all the constraints of the entry pairs in $\Omega_1, ..., \Omega_M$, as well as the constraints in Property 3.1, we can formulate the problem to derive the optimal rating policy for all requesters.

However, due to the diverse task features as well as requesters' different focus on cost, work quality, and rating, the feasible region restricted by the constraints from $\Omega_1, ..., \Omega_M$ is possibly an empty set, indicating that it is hard to find a universal rating policy that can align all the requesters to optimize the platform utility. As a solution, we relax the problem by introducing each constraint $z_{I_j^*} - z_I \geq \psi_{I_j^*,I}$ into the objective function:

$$\min \sum_j \sum_{(I,I_j^*)\in\Omega_j} \Lambda\left(z_I + \psi_{I_j^*,I} - z_{I_j^*}\right) \tag{21}$$

where Λ is defined as

$$\Lambda(w) = \begin{cases} w & w \geq 0 \\ 0 & \text{otherwise} \end{cases}. \tag{22}$$

meaning that the objective function get penalized if any constraints are violated.

Here, we introduce an intermediate variable $t_{I_j^*,I} \geq 0$ for each constraint $z_{I_j^*} - z_I \geq \psi_{I_j^*,I}$, with

$$-z_{I_j^*} + z_I + \psi_{I_j^*,I} - t_{I_j^*,I} \leq 0 \tag{23}$$

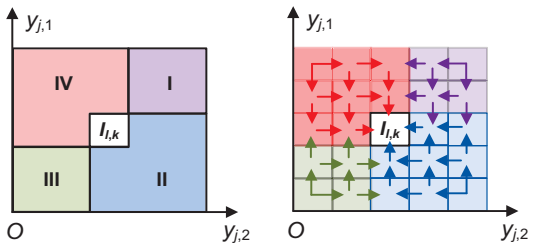
Then, the objective function defined in Equation (21) can be rewritten as

$$\min \quad \sum_j \sum_{(I,I_j^*)\in\Omega_j} t_{I_j^*,I} \tag{24}$$

$$\text{s.t.} \quad t_{I_j^*,I} \geq 0, \forall (I, I_j^*) \in \Omega_j, j = 1, ..., M. \tag{25}$$

By additionally considering the constraints in Property 3.1, the above problem (in Equations (24)(25)) is essentially a linear programming (LP) problem that can be solved by existing approaches such as the simplex methods [17].

Finally, for theoretical interests, we also derive an upper bound of the platform utility in Proposition 3.2:



(a) Partition regions (b) Flow directions in each region

Figure 2: Partitioned regions in $\mathcal{Y}$ to ensure $\mathbf{y}_j$ to flow to $l_{l,k}$.

Proposition 3.2. *$\sum_j g_j\left(\mathbf{y}_j^*; \mathbf{B}, \mathbf{e}_j\right)$ offers an upper bound of the platform utility in ORP.*

Proof. The detailed proof can be found in Appendix. □

By comparing this theoretical bound with the platform utility obtained by our heuristic algorithm, we can check how close our approach achieves the optimal. The detailed comparison between our solution and this upper bound will be given in Section 4.

4 PERFORMANCE EVALUATION

We first validate the central hypothesis in our model via a real user study in Section 4.1. After that, we conduct simulations to evaluate the performance of our rating policy based on both real trace data and synthetic data in Section 4.2.

4.1 MTurk User Study

We conduct a controlled experiment with users on MTurk (approved through institutional IRB No. 00012140) to validate the main assumption of the model, namely, the power of an appropriately-designed rating policy to impact requesters' willingness to support platform objectives. In this case specifically, we target the inclusion of contributors with less experience.

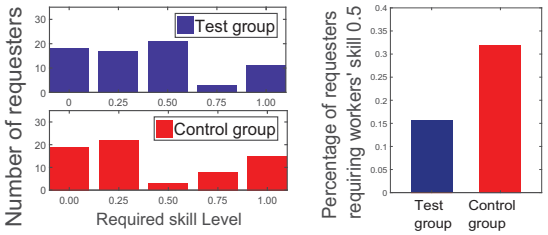
Study Design. The task we deploy asks participants to serve as task requesters, designing a task to post on Turk. We tell participants that we will post their tasks to Turk to be completed by Turk contributors, and that they will receive a bonus based on their task's real outcomes (varied for Control and Test groups, see below).

Our instructions are as follows: *Your contributors have to label* 50 *tweets. Each Tweet needs to be labeled as including (or not) evidence of self-disclosure (i.e., revelation of personal information). Each Tweet should be labelled by* 3 *unique contributors, for a total of* 150 *labels.*

Participants are asked to select the number of individual tweets to be assigned per task, and the corresponding payment policy. They are not given a set budget, but are asked to imagine that the money is their own. Participants are additionally asked to select inclusion criteria for their contributors. They may select from the following inclusion criteria:

- ▷ Master-Only
- ▷ Specific Skills (select any combination of the following)
  - English language speaker
  - Location within the United States
  - Social media user
- ▷ No Prerequisites

The experimental manipulation is contained within the contributors' "bonus". That is, the Control and Test groups are given the



(a) Histograms of skill requirement (b) Comparison for skill ≥ 0.5

Figure 3: Comparison of required skill levels between the test group and the control group

following information, respectively, about distribution of bonuses to study participants.

1) *Control group*: *You will receive a bonus of up to $0.50 based on the quality of responses we receive, and an additional $0.50 based on your* ***rating****. In turn, your rating is generated* 50% *by your contributors (presumably based on fair pay) and* 50% *by the platform based on your willingness to engage less experienced users.*
2) *Test group*: *You will receive a bonus of up to $1 based on the quality of responses we receive.*

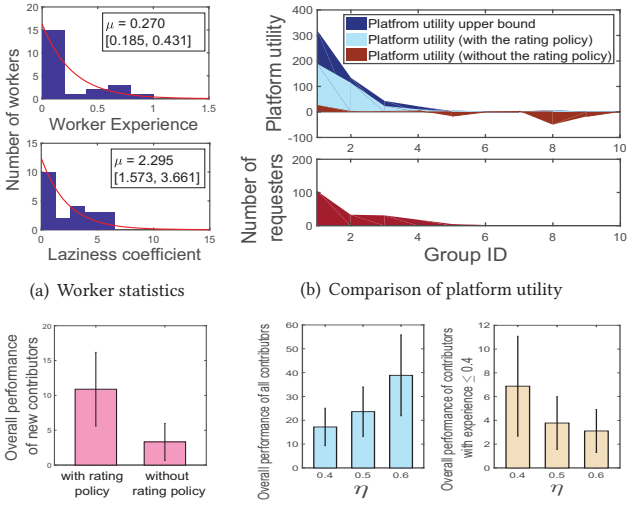
Findings. We collected responses from 140 participants, 70 Control and 70 Test. After discarding 3 responses that did not pass our quality check, we remained with 67 participants in the Control condition and 70 in the Test condition.

We are primarily interested in exploring the impact of our experimental manipulation, namely, a rating scheme that explicitly rewards inclusion of less experienced contributors. At a fundamental level, if requesters are evaluated based on their willingness to adhere to platform objectives, will that have an impact on their behavior?

We first normalize inclusion criteria on the interval [0, 1]. Selection of *No prerequisites* maps to 0, *Master-Only* maps to 1, and selection of any subset of 1, 2, or 3 *Selected Skills* maps to 0.25, 0.5 and 0.75, respectively. We find no significant difference in the mean skill level requested by Control and Experimental groups (0.42 and 0.4, respectively). The task as we proposed it motivates some basic inclusion criteria, e.g., English language proficiency. However, we also observe that the rating scheme we present does serve to reduce the likelihood requesters will restrict their inclusion criteria to *Master-Only*, or even a requirement of all three selected skills. Figure 3(a) gives the distributions of skill levels selected by participants in the Control and Test groups. Figure 3(b) compares the percentages of requesters setting inclusion criteria greater than or equal to 0.5. In our experiment, this translated to at least two selected skills, or Master-Only status. More generally, we envision this bound to represent a divide between skills that are absolutely necessary to credibly complete the task, and skills that are "nice to have" but may be overly exclusionary for new or less experienced contributors in the platform [30].

4.2 Simulation

Simulation based on real trace. We carry out an extensive evaluation of our proposed rating policy using a real dataset (over 3 million task records) extracted from MTurk from September 2014 to



(a) Worker statistics (b) Comparison of platform utility

(c) New contributors' performance with/without rating (d) Overall performance of all contributors and contributors with experience ≤ 0.4

Figure 4: Simulation based on real dataset.

January 2017 [15]. This data includes contributor identifiers, qualification (experience), partial records (29.6% of *human intelligence tasks (HITs)*) of when a contributor progresses through the different stages of completing HITs (including *accept, submit, abandon*), a partial record of subsequent requester actions (including *accept, reject*), and requesters' payments to contributors. We here normalize contributors' experience to interval [0, 1] and set the new contributor threshold $\eta = 0.5$ by default, i.e., contributors with experience lower than 0.5 are considered as "new" contributors. Later, we also change the value of η and check how the performance of the rating policy is impacted (in Figure 5(d)).

In the following, we test three metrics:
1) the *platform utility* (defined by Equation (12));
2) the *overall performance of new contributors*, defined as the sum performance of all new contributors, i.e., $\sum_{b_{i,1} \le \eta} \xi\left(\phi\left(\mathbf{y}_j; \alpha_i^{\text{laz}}\right), \mathbf{b}_i, \mathbf{e}_j\right)$.
3) the *approximation ratio* of the platform utility, defined as the ratio of platform utility's upper bound (derived in *Proposition 3.2*) to the utility obtained by our rating policy.

We first filter out the requesters and contributors with less than 10 HITs. Then, for the first experiment, we randomly select a representative sample of 22 contributors (with 7994 submitted HITs). We consider contributors' complete ratio and approved ratio as proxies of their effort level and performance, respectively. According to Equation (15), each contributor i's reaction $x_{i,j}$ has a linear relationship with $1/y_{j,1}$ ($y_{j,1}$ denotes requester j's experience requirement). Then, given a group of the contributor's behavior records and the corresponding requirements from requesters, we estimate the contributor's laziness coefficient by using linear regression [38]. Figure 4(a) depicts the histograms of contributors' laziness coefficient α_i^{laz}, from which we observe that around 45.5% contributors have α_i^{laz} lower than 1. Figure 4(a) also shows the distribution of contributors' experience, where 81.8% contributors are identified as new contributors, i.e., with experience lower than 0.5.

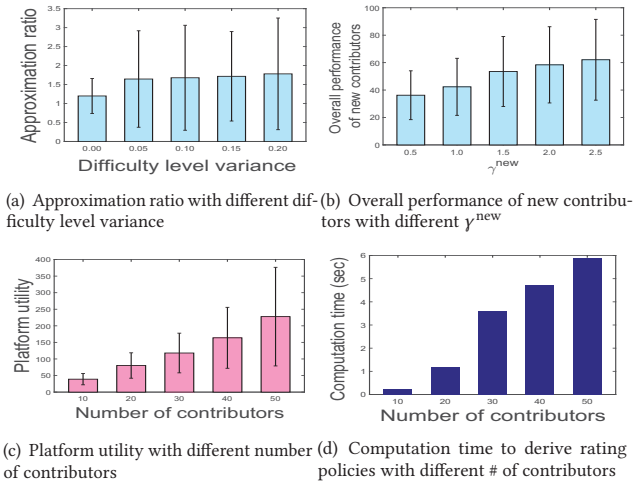
We compare the platform utility with/without the rating policy and its upper bound in Figure 4(b). Here, we categorize all HITs

into 10 different groups according to the task required qualification (such as "Adult Content qualification", "Native Hindi speaker"), and sort these group based on the number of requesters participated in. We note that the rating may impact or change both requesters and contributors' decisions compared with their actual recorded behavior. Hence, in the simulation with the rating policy, we derive the requesters and the contributors' behavior based on the assumption that they always maximize their own utilities (defined by Equations (4) and (3), respectively). Figure 4(b) demonstrates that the platform utility is significantly improved with the rating policy, which effectively motivates requesters to get new contributors involved as possible. Using our rating policy, the average approximation ratio of the platform utility across the 10 groups is 1.18, indicating that our approach can achieve the optimal closely. Figure 4(c) compares the new contributors' overall performance with and without the rating policy. Not surprisingly, motivated by our rating policy, the performance contributed by new contributors is much higher (214.3%) than that with no rating policy.

In addition, Figure 4(d) shows the platform utility achieved by our rating policy with η increased from 0.4 to 0.5, where η denotes the threshold to identify "new contributors". As we observed, to ensure high quality work from contributors while still securing fair rating, requesters are more likely to hire contributors of which the experience is close to or slightly lower than η. Hence, when the platform looses the criteria for identifying "new" contributors, it actually offers requesters more space to hire more experienced contributors, leading to a higher overall performance from all the contributors. However, higher η may discourage participation from new contributors whose experience is far below η. As the right sub-figure in Figure 4(d) shows, when η is changed from 0.4 to 0.5 or 0.6, the overall performance of contributors with experience ≤ 0.4 is decreased, since requesters prefer to choose contributors with experience in level 5 or 6 rather than in the levels lower than 4 in this case. This experimental result also indicates the importance to identify the targeted contributor group to incentivize when determining the threshold η.

Simulation based on synthetic data. We next evaluate the performance of our rating policy via synthetically generated data. The main parameters in the synthetic data, including contributors' laziness coefficient and experience, and requesters' cost and rating coefficients still follow the same distribution of the original dataset [15].

As our rating policy has to deal with conflicting objectives from multiple requesters, it is of great interest to check how the variance of task features from different requesters can impact the rating policy's performance. Here, we consider task difficulty level as its main feature, which is reflected by the ratio of contributors' overall performance on this task to the contributors' effort level. We generate 5 groups of tasks with their difficulty levels following normal distribution, with the mean values all equal to 0.417 (same as it is in the dataset [15]) and the standard deviation increased from 0.0 to 0.20. We depict the platform utility's approximation ratio given these 5 group tasks in Figure 5(a). As shown in the figure, the approximation ratio increases with the increase of the difficulty level standard deviation, implying bigger gap between our solution and the upper bound when the task variance is higher. This is because that, with higher diverse task features across different



(a) Approximation ratio with different difficulty level variance

(b) Overall performance of new contributors with different γ^{new}

(c) Platform utility with different number of contributors

(d) Computation time to derive rating policies with different # of contributors

Figure 5: Simulation based on synthetic data.

requesters, requesters are more likely to have different focus on task quality, cost, and rating, making it more difficult for our algorithm to find a universal rating policy to align all requesters to optimize the platform utility.

Figure 5(b) shows the overall performance from contributors with the coefficient γ^{new} increased from 0.5 to 2.5, where γ^{new} defined in the platform utility (Equation (12)) implies how much the platform focuses on hiring new contributors. As expected, new contributors' performance improves when the platform assigns higher weights to new contributors' participation in its utility.

Finally, in Figure 5(c), we compare the platform utility with an increasing number of contributors, from 10 to 50. To ensure that the newly generated contributors are statistically similar to the real dataset [15], we first fit the distribution of contributors' experience and laziness coefficient in the real dataset with exponential curves: $0.270e^{-0.270}$ and $2.295e^{-2.295}$, respectively, as shown in Figure 4(a). Then, we generate contributors with their experience and laziness coefficient following these two exponential distributions. The figure demonstrates that higher number of contributors create higher utility for the platform, indicating the importance of attracting new contributors to participate in crowdsourcing services. In addition, computational cost is also a concern when increasing the size of contributor pool. Hence, we compare the computation time to derive the rating policy with the different number of contributors in Figure 5(d). Even though the computation time increases with the increase of number of contributors in the figure, the computation time with 50 contributors is still lower than 6 seconds, which is still acceptable.

5 RELATED WORK

Recent work has brought to light some of the critical challenges of developing safe, ethical and effective crowdsourcing systems [3, 19, 22, 33]. While some crowdsourcing systems consistently attract high-quality contributors, other seemingly similar ones suffer from low quality work, or even fail due to too little participation [9]. As contributors in crowdsourcing are mostly self-centered, it is of great importance to understand how to provide incentives for contributors to provide high-quality work, which has been studied for decades by a variety of work from different domains [10, 12, 18].

The types of reward (or compensation) used as incentive for contributors varies depending on the particular application. Some crowdsourcing systems (e.g., MTurk [6]) offer financial incentives for participation [18], many others are driven by social-psychological rewards, e.g., both intrinsic motivators like interest [10, 12, 20] or the satisfaction of benefiting a cause [9] (e.g., participating in a scientific research [13, 40]), as well as extrinsic social rewards such as reputation or status [1]. There is now a growing effort in social psychology that aims to address what motivates contributors in crowdsourcing systems [2, 21, 23, 24]. A related and well studied research problem is *how to allocate rewards to incentivize desirable outcomes* [18]. For example, in online knowledge sharing forums (e.g., Quora), efforts have been devoted to studying reward allocation issues, including what reward policies elicit quicker responses from contributors [20], how to distribute attention rewards amongst contributors [12], as well as how to reward contributors regarding the implementability of outcomes [10].

Game theoretic approaches. Many incentive strategies are based on the analysis of users' reaction and interaction, resorting to *game theoretical models*. A game-theoretic approach to incentive design, in general, proceeds by constructing an appropriate model where users make decisions over an action space that are typically associated with users' benefits and costs [9, 11]. Generally, there are two research directions amongst game-theoretic approaches to incentives: 1) analyze users' behavior equilibrium under given compensation policies to predict users' reaction (e.g., [4, 44]), and 2) build the compensation policies (i.e., which rewards are allocated) to motivate user behavior that achieves some particular objectives (e.g., [11, 37]). While a game-theoretic approach has the general structure described above, each application comes with its own unique features, depending on the common knowledge shared amongst different stakeholders, as well as the nature of different players' strategy space. For example, game with a purpose (GWAP) stands as a widely used game theoretical model for crowdsourcing incentivization, where users who are ostensibly simply playing the game also simultaneously produce useful input to a computation or task [9]. This framework has been applied to many applications like ESP game [25], but it is limited to scenarios where contributors are intrinsically driven. A more interesting approach, closer to the framework we investigate here, follows hierarchical leader-follower structures, where contributors are extrinsically driven by awards offered by requesters [29, 37]. Stackelberg schemes are a natural choice for modeling this leader-follower structure wherein a task requester (leader) typically outlines the work to be done and compensation to be awarded, and contributors (followers) are required to respond within those parameters (even if that response is not to engage with the task) [43].

Finally, Gaikwad and colleagues [8] have explored a new approach to incentive compatibility for more accurate ratings in crowdsourcing platforms. This work is complementary to ours, but focuses more narrowly on socio-technical solutions to the specific problem of mediating reputation inflation in crowdsourcing systems, and does not provide a theoretical framework with which to reason about compensation policies and platform sustainability.

Notably, most studies to date consider incentive mechanisms solely from the perspective of requester, i.e., aim to improve of contributors' quality of work, but neglect the benefits of contributors. Central to this proposed effort is our assertion that these asymmetric approaches are no longer sufficient, compromise platform health, and undermine important opportunities for broader social benefit.

6 CONCLUSIONS

In this paper, we have constructed a game-theoretic representation of the interactions amongst contributors, requesters, and the platform itself in crowdsourcing environments. Toward the development of formal mechanisms to improve overall platform sustainability and to ameliorate the prototypical imbalance between contributors and requesters in these environments, we have proposed a rating algorithm for task requesters. We have validated the premise and efficacy of this algorithm through controlled user experiments and through comprehensive simulations based on massive-scale data derived from MTurk. We envision this work as a foundational step in creating more inclusive and sustainable crowdsourcing platforms.

The framework we propose is sufficiently general to support further specification of contributor, requester and platform objectives. For example, we solicited qualitative feedback from participants in the user study (see Section 4.1) about best practices for designing good tasks. That is, we asked them what requesters might do to improve the quality and attractiveness of their tasks. Overwhelmingly, responses focused on issues of task clarity: make instructions simple, make instructions clear, provide examples. This feedback is an important pointer for future iterations of contributors' objective function development, in particular in the specific context of microtasking platforms. We focus on payment policy as the primary input to contributor satisfaction, but we might in the future explicitly consider clarity of task as an independent feature.

ACKNOWLEDGMENTS

This work was partly supported by the National Science Foundation under Grant 1453080.

REFERENCES

[1] Mohammad Allahbakhsh, Aleksandar IgnjatoviÄĞ, Boualem Benatallah, Amin Beheshti, Elisa Bertino, and Norman Foo. 2012. Reputation Management in Crowdsourcing Systems. 664–671. https://doi.org/10.4108/icst.collaboratecom.2012.250499

[2] Harish Arelli and Francesca Spezzano. 2017. Who Will Stop Contributing?: Predicting Inactive Editors in Wikipedia. In *ASONAM*.

[3] Benjamin B Bederson and Alexander J Quinn. 2011. Web workers unite! addressing challenges of online laborers. In *CHI 2011 EA*. ACM, 97–106.

[4] Vincent Conitzer. 2012. Prediction Markets, Mechanism Design, and Cooperative Game Theory. *CoRR* abs/1205.2654 (2012). arXiv:1205.2654 http://arxiv.org/abs/1205.2654

[5] S. Dempe, J. Dutta, and B. S. Mordukhovich. 2007. New necessary optimality conditions in optimistic bilevel programming. *Optimization* 56, 5-6 (2007), 577–604. https://doi.org/10.1080/02331930701617551 arXiv:https://doi.org/10.1080/02331930701617551

[6] Xiaotie Deng. 1998. *Complexity Issues in Bilevel Linear Programming*. Springer US, Boston, MA, 149–164. https://doi.org/10.1007/978-1-4613-0307-7_6

[7] Francisco J. Candido et al. 2015. Crowdsourcing the General Public for Large Scale Molecular Pathology Studies in Cancer. *EBioMedicine* 2, 7 (2015), 681 – 689. https://doi.org/10.1016/j.ebiom.2015.05.009

[8] Snehalkumar (Neil) S. et al. Gaikwad. 2016. Boomerang: Rebounding the Consequences of Reputation Feedback on Crowdsourcing Platforms. In *Proc. of the 29th Annual Symposium on User Interface Software and Technology (UIST '16)*. ACM, New York, NY, USA, 625–637. https://doi.org/10.1145/2984511.2984542

[9] Arpita Ghosh. 2013. *Game Theory and Incentives in Human Computation Systems*. Springer New York, New York, NY, 725–742. https://doi.org/10.1007/978-1-4614-8806-4_58

[10] Arpita Ghosh and Patrick Hummel. 2012. Implementing Optimal Outcomes in Social Computing: A Game-theoretic Approach. In *Proc. of the 21st International*

Conference on World Wide Web (WWW '12). ACM, New York, NY, USA, 539–548. https://doi.org/10.1145/2187836.2187910
[11] Arpita Ghosh and Patrick Hummel. 2012. Implementing Optimal Outcomes in Social Computing: A Game-theoretic Approach. In *Proc. of the 21st International Conference on World Wide Web (WWW '12)*. ACM, New York, NY, USA, 539–548. https://doi.org/10.1145/2187836.2187910
[12] Arpita Ghosh and Preston McAfee. 2012. Crowdsourcing with Endogenous Entry. In *Proc. of the 21st International Conference on World Wide Web (WWW '12)*. ACM, New York, NY, USA, 999–1008. https://doi.org/10.1145/2187836.2187970
[13] U.S. government. 2019. Federal Crowdsourcing and Citizen Science Toolkit. https://www.citizenscience.gov/toolkit/#/. (2019). [Online; accessed 13-January-2019].
[14] Pierre Hansen, Brigitte Jaumard, and Gilles Savard. 1992. New Branch-and-bound Rules for Linear Bilevel Programming. *SIAM J. Sci. Stat. Comput.* 13, 5 (Sept. 1992), 1194–1217. https://doi.org/10.1137/0913069
[15] Kotaro Hara, Abigail Adams, Kristy Milland, Saiph Savage, Chris Callison-Burch, and Jeffrey P. Bigham. 2018. A Data-Driven Analysis of Workers' Earnings on Amazon Mechanical Turk. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, 449:1–449:14.
[16] Patrick T Harker and Jong-Shi Pang. 1988. Existence of optimal solutions to mathematical programs with equilibrium constraints. *Operations Research Letters* 7, 2 (1988), 61 – 64. https://doi.org/10.1016/0167-6377(88)90066-1
[17] Frederick S. Hillier. 2008. *Linear and Nonlinear Programming*. Stanford University.
[18] Zehong Hu and Jie Zhang. 2017. Optimal posted-price mechanism in microtask crowdsourcing.. In *Proc. of AAAI*.
[19] Lilly Irani. 2015. Difference and Dependence among Digital Workers: The Case of Amazon Mechanical Turk. *South Atlantic Quarterly* 114, 1 (2015), 225–234.
[20] Shaili Jain, Yiling Chen, and David C. Parkes. 2009. Designing Incentives for Online Question and Answer Forums. In *Proc. of the 10th ACM Conference on Electronic Commerce (EC '09)*. ACM, New York, NY, USA, 129–138. https://doi.org/10.1145/1566374.1566393
[21] Lian Jian and Jeffrey K. MacKie-Mason. 2012. Incentive-Centered Design for User-Contributed Content.
[22] Aniket Kittur, Jeffrey V Nickerson, Michael Bernstein, Elizabeth Gerber, Aaron Shaw, John Zimmerman, Matt Lease, and John Horton. 2013. The future of crowd work. In *Proc. of CSCW 2013*. ACM, 1301–1318.
[23] Sanjay Krishnan, Jay Patel, Michael J. Franklin, Ken Goldberg, and patel. jay. 2014. Social Influence Bias in Recommender Systems : A Methodology for Learning , Analyzing , and Mitigating Bias in Ratings.
[24] Sanjay Krishnan, Jay Patel, Michael J. Franklin, and Kenneth Y. Goldberg. 2014. A methodology for learning, analyzing, and mitigating social influence bias in recommender systems. In *RecSys*.
[25] ESP Lab. 2019. ESP Gaming. https://www.espgaming.com/. (2019). [Online; accessed 13-January-2019].
[26] David C. Lay, Steven R. Lay, and Judi J. McDonald. 2015. . Pearson.
[27] Maria Beatrice Lignola and Jacqueline Morgan. 2001. *Existence of Solutions to Bilevel Variational Problems in Banach Spaces*. Springer US, Boston, MA, 161–174. https://doi.org/10.1007/0-306-48026-3_10
[28] Ioana Literat. 2012. The Work of Art in the Age of Mediated Participation: Crowdsourced Art and Collective Creativity. *International Journal of Communication* 6 (01 2012), 2962–2984.
[29] Shuyun Luo, Yongmei Sun, Yuefeng Ji, and Dong Zhao. 2015. Stackelberg Game Based Incentive Mechanisms for Multiple Collaborative Tasks in Mobile Crowdsourcing. *Mobile Networks and Applications* (12 2015). https://doi.org/10.1007/s11036-015-0659-3
[30] David Martin, Benjamin V. Hanrahan, Jacki O'Neill, and Neha Gupta. 2014. Being a Turker. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW '14)*. ACM, New York, NY, USA, 224–235. https://doi.org/10.1145/2531602.2531663
[31] David Martin, Benjamin V. Hanrahan, Jacki O'Neill, and Neha Gupta. 2014. Being A Turker. *Proc. of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing* (2 2014), 224–235. https://doi.org/10.1145/2531602.2531663
[32] Andreas Mladenow, Christine Bauer, and Christine Strauss. 2014. Social Crowd Integration in New Product Development: Crowdsourcing Communities Nourish the Open Innovation Paradigm. *Global Journal of Flexible Systems Management* 15, 1 (01 Mar 2014), 77–86. https://doi.org/10.1007/s40171-013-0053-6
[33] Jacki O'Neill and David Martin. 2013. Relationship-Based Business Process Crowdsourcing. In *INTERACT '13*. Springer Berlin Heidelberg, 429–446.
[34] J. V. Outrata. 1993. Necessary optimality conditions for Stackelberg problems. *Journal of Optimization Theory and Applications* 76, 2 (01 Feb 1993), 305–320. https://doi.org/10.1007/BF00939610
[35] C. Qiu, A. Squicciarini, D. R. Khare, B. Carminati, and J. Caverlee. 2018. CrowdEval: A Cost-Efficient Strategy to Evaluate Crowdsourced Worker's Reliability. In *Proc. of the International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*. Richland, SC, 1486–1494.
[36] C. Qiu, A. C. Squicciarini, and Ben Hanrahan. 2019. Incentivizing Distributive Fairness for Crowdsourcing Workers. In *Proc. of International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.
[37] C. Qiu, A. C. Squicciarini, S. M. Rajtmajer, and J. Caverlee. 2017. Dynamic Contract Design for Heterogenous Workers in Crowdsourcing for Quality Control. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*. 1168–1177. https://doi.org/10.1109/ICDCS.2017.187
[38] S. M. Ross. 2003. *Introduction to Probability Models, 8th Edition*. Amsterdam: Academic Press.
[39] Gregory D. Saxton, Onook Oh, and Rajiv Kishore. 2013. Rules of Crowdsourcing: Models, Issues, and Systems of Control. *Information Systems Management* 30, 1 (2013), 2–20. https://doi.org/10.1080/10580530.2013.739883 arXiv:https://doi.org/10.1080/10580530.2013.739883
[40] Jennifer L. Shirk, Heidi L. Ballard, Candie C. Wilderman, Tina Phillips, Andrea Wiggins, Rebecca Jordan, Ellen McCallie, Matthew Minarchek, Bruce V. Lewenstein, Marianne E. Krasny, and Rick Bonney. 2012. Public Participation in Scientific Research: a Framework for Deliberate Design. *Ecology and Society* 17, 2 (2012). http://www.jstor.org/stable/26269051
[41] A. Sinha, P. Malo, and K. Deb. 2018. A Review on Bilevel Optimization: From Classical to Evolutionary Approaches and Applications. *IEEE Transactions on Evolutionary Computation* 22, 2 (April 2018), 276–295. https://doi.org/10.1109/TEVC.2017.2712906
[42] L. Vicente, G. Savard, and J. Júdice. 1994. Descent approaches for quadratic bilevel programming. *Journal of Optimization Theory and Applications* 81, 2 (01 May 1994), 379–399. https://doi.org/10.1007/BF02191670
[43] Andrzej WilczyÅĎski, Agnieszka JakÃąbik, and Joanna KoÅĆodziej. 2016. Stackelberg Security Games: Models, Applications and Computational Aspects. *Journal of Telecommunications and Information Technology* 3 (09 2016), 70–79.
[44] Q. Zhu, C. Fung, R. Boutaba, and T. Basar. 2009. A game-theoretical approach to incentive design in collaborative intrusion detection networks. In *2009 International Conference on Game Theory for Networks*. 384–392. https://doi.org/10.1109/GAMENETS.2009.5137424

.1 Proof of Proposition 3.1

Proof. According to the compensation function described by Equation (10) and Equation (11), for each contributor i:

Case I: If $e_{i,j} < y_{j,2}$ (i.e., the contributor's experience is lower than the requirement from the requester j), contributor i's submitted answers won't be accepted by requester j. In this case, the contributor's utility function is derived as $-\alpha_i^{\text{laz}} x_{i,j}$, indicating that the contributor's best response will be $x_{i,j}^* = 0$.

Case II: If $e_{i,j} \geq y_{j,2}$ (i.e., the contributor's experience satisfies requester j's requirement), contributor i's submitted answers will be accepted by the requester. In this case, contributor i's utility is derived as a quadratic function

$$f\left(x_{i,j}; \mathbf{y}_j\right) - \alpha_i^{\text{laz}} x_{i,j} = \left(2y_{j,1} - \alpha_i^{\text{laz}}\right) x_{i,j} - y_{j,1} x_{i,j}^2 \tag{26}$$

of which the optimal $x_{i,j}^*$ can be obtained $x_{i,j}^* = \frac{2\mu y_{j,1} - \alpha_i^{\text{laz}}}{y_{j,1}}$. Note that the effort level $x_{i,j}$ is normalized to $[0, 1]$, and hence we have

$$x_{i,j} = \begin{cases} 0 & \frac{2y_{j,1} - \alpha_i^{\text{laz}}}{y_{j,1}} < 0 \\ \frac{2y_{j,1} - \alpha_i^{\text{laz}}}{y_{j,1}} & 0 \leq y_{j,1} \leq 1 \\ 1 & \frac{2y_{j,1} - \alpha_i^{\text{laz}}}{y_{j,1}} > 1 \end{cases} \tag{27}$$

where $\frac{2y_{j,1} - \alpha_i^{\text{laz}}}{y_{j,1}} < 0 \Rightarrow y_{j,1} \in \left[0, \frac{\alpha_i^{\text{laz}}}{2}\right)$, $0 \leq y_{j,1} \leq 1 \Rightarrow y_{j,1} \in \left[\frac{\alpha_i^{\text{laz}}}{2}, \alpha_i^{\text{laz}}\right]$, and $\frac{2y_{j,1} - \alpha_i^{\text{laz}}}{y_{j,1}} > 1 \Rightarrow y_{j,1} \in \left(\alpha_i^{\text{laz}}, \infty\right)$. □

.2 Proof of Proposition 3.2

Proof. Let $\overline{\mathbf{Z}}$ denote the optimal rating policy for ORP, under which the optimal decisions made by requesters are $\overline{\mathbf{Y}} = [\overline{\mathbf{y}}_1, ..., \overline{\mathbf{y}}_M]$. Then, the maximum platform utility is given by $g\left(\mathbf{X}, \overline{\mathbf{Y}}; \mathbf{B}, \mathbf{E}\right) = \sum_j g_j\left(\overline{\mathbf{y}}_j; \mathbf{B}, \mathbf{e}_j\right)$. According to Equation (17), $g_j\left(\mathbf{y}_j^*; \mathbf{B}, \mathbf{e}_j\right) \geq g_j\left(\overline{\mathbf{y}}_j; \mathbf{B}, \mathbf{e}_j\right)$, and hence $g\left(\mathbf{X}, \overline{\mathbf{Y}}; \mathbf{B}, \mathbf{E}\right) \leq \sum_j g_j\left(\mathbf{y}_j^*; \mathbf{B}, \mathbf{e}_j\right)$. □